|  |  |
| --- | --- |
|  | Министерство науки и высшего образования Российской Федерации  Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»  (МГТУ им. Н.Э. Баумана) |
| ФАКУЛЬТЕТ «ИНЖЕНЕРНЫЙ БИЗНЕС И МЕНЕДЖМЕНТ»  КАФЕДРА «ПРОМЫШЛЕННАЯ ЛОГИСТИКА» (ИБМ-3)  Лабораторная работа №1  По дисциплине:  «Парадигмы и конструкции языков программирования»  Студент ИБМ3- 34Б П. Я. Головастикова  (Подпись, дата)    Руководитель Ю. Е. Гапанюк  (Подпись, дата)  2025 г. | |

**Основные конструкции языка Python.**

Задание: разработать программу для решения биквардратного уравнения.

Реализация на языке python с помощью процедурной парадигмы:

import math  
import sys  
  
def getnumber(prompt):  
 while True:  
 try:  
 value = float(input(prompt))  
 return value  
 except ValueError:  
 print("Введите корректное число")  
  
  
def solve(a, b, c):  
 if a == 0:  
 print("Нет корней")  
 return []  
 roots = []  
 D = b \*\* 2 - 4 \* a \* c  
 if D < 0:  
 return []  
 elif D == 0:  
 roots.append(-b / (2.0 \* a))  
 elif D > 0:  
 sqd = math.sqrt(D)  
 roots.extend([(-b + sqd) / (2.0 \* a), (-b - sqd) / (2.0 \* a)])  
 return roots  
  
  
def main():  
 coef = []  
  
 if len(sys.argv) >= 4:  
 try:  
 coef = [float(sys.argv[1]), float(sys.argv[2]), float(sys.argv[3])]  
 print(f"Коэффициенты из командной строки: A={coef[0]}, B={coef[1]}, C={coef[2]}")  
 except ValueError:  
 print("Некорректные аргументы командной строки. Ввод с клавиатуры:")  
 coef = []  
  
 if len(coef) != 3:  
 a = getnumber("Введите коэффициент A: ")  
 b = getnumber("Введите коэффициент B: ")  
 c = getnumber("Введите коэффициент C: ")  
 else:  
 a, b, c = coef  
  
 print(f"\nРешаем уравнение: {a}\*x^2 + {b}\*2\*x + {c} = 0")  
  
 roots = solve(a, b, c)  
  
 if roots:  
 print(f"Действительные корни: {sorted(roots)}")  
 else:  
 print("Действительных корней нет")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Результат выполнения программы:

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Реализация на языке python на основе объекто-ориентированной парадигмы:

import sys  
import math  
  
  
class Biquadrat:  
 def \_\_init\_\_(self, a=0, b=0, c=0):  
 self.a = a  
 self.b = b  
 self.c = c  
  
 def solve(self):  
 res = []  
 D = self.b \*\* 2 - 4 \* self.a \* self.c  
 if D < 0:  
 return []  
 elif D == 0:  
 res.append(-self.b / (2.0 \* self.a))  
 elif D > 0:  
 sqd = math.sqrt(D)  
 res.extend([(-self.b + sqd) / (2.0 \* self.a), (-self.b - sqd) / (2.0 \* self.a)])  
 return sorted(res)  
  
  
class Input:  
 @staticmethod  
 def coef(index, prompt):  
 while True:  
 try:  
 if index < len(sys.argv):  
 coef\_str = sys.argv[index]  
 else:  
 print(prompt)  
 coef\_str = input()  
  
 coeff = float(coef\_str)  
 return coeff  
 except (ValueError, IndexError):  
 print("Введите корректное число.")  
  
  
def main():  
 a = Input.coef(1, "Введите коэффициент A: ")  
 b = Input.coef(2, "Введите коэффициент B: ")  
 c = Input.coef(3, "Введите коэффициент C: ")  
  
 equation = Biquadrat(a, b, c)  
 roots = equation.solve()  
 if roots:  
 print(f"Действительные корни: {roots}")  
 else:  
 print("Действительных корней нет")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Результат выполнения программы:

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Реализация на языке Java:

import java.util.Scanner;  
import java.util.ArrayList;  
import java.util.Collections;  
  
public class Biquadrat {  
 private double a;  
 private double b;  
 private double c;  
  
 public Biquadrat(double a, double b, double c) {  
 this.a = a;  
 this.b = b;  
 this.c = c;  
 }  
  
 public ArrayList < Double > solve() {  
 ArrayList < Double > roots = new ArrayList <> ();  
 double D = b \* b - 4 \* a \* c;  
 double sqd=Math.sqrt(D);  
 if (D < 0) {  
 return roots;  
 }  
 if (D == 0){  
 roots.add((-b / (2.0 \* a)));  
 }  
 if (D > 0){  
 roots.add((-b + sqd) / (2.0 \* a));  
 roots.add((-b - sqd) / (2.0 \* a));  
 }  
 Collections.sort(roots);  
 return roots;  
 }  
  
@Override  
public String toString() {  
 return String.format("%.2f\*x^4 + %.2f\*x^2 + %.2f = 0", a, b, c);  
}  
  
public static double getValidDouble(Scanner scanner, String prompt) {  
 while (true) {  
 System.out.print(prompt);  
 try {  
 return Double.parseDouble(scanner.nextLine());  
 } catch(NumberFormatExceptione) {  
 System.out.println("Ошибка! Введите корректное число.");  
 }  
 }  
}  
  
public static void main(String[] args) {  
 Scanner scanner = new Scanner(System. in);  
 double a = 0, b = 0, c = 0;  
  
 if (args.length >= 3)  
 {  
 try {  
 a = Double.parseDouble(args[0]);  
 b = Double.parseDouble(args[1]);  
 c = Double.parseDouble(args[2]);  
 System.out.printf("Коэффициенты из командной строки: A=%.2f, B=%.2f, C=%.2f%n", a, b, c);  
 } catch (NumberFormatException e) {  
 System.out.println("Некорректные аргументы командной строки. Ввод с клавиатуры:");  
 a = b = c = 0;  
 }  
 }  
 if (a == 0) {  
 while (true) {  
 try {  
 a = getValidDouble(scanner, "Введите коэффициент A: ");  
 b = getValidDouble(scanner, "Введите коэффициент B: ");  
 c = getValidDouble(scanner, "Введите коэффициент C: ");  
 break;  
 } catch(IllegalArgumentExceptione) {  
 System.out.println(e.getMessage());  
 }  
 }  
 }  
  
 try {  
 Biquadrat equation = new Biquadrat(a, b, c);  
 System.out.printf("%nРешаем уравнение: %s%n", equation);  
  
 ArrayList < Double > roots = equation.solve();  
  
 if (roots.isEmpty()) {  
 System.out.println("Действительных корней нет");  
 } else {  
 System.out.print("Действительные корни: ");  
 for (int i = 0; i < roots.size(); i++) {  
 if (i > 0) System.out.print(", ");  
 System.out.printf("%.4f", roots.get(i));  
 }  
 System.out.println();  
 }  
 } catch (IllegalArgumentException e) {  
 System.out.println("Ошибка: " + e.getMessage());  
 }  
  
scanner.close();  
}  
}

Результат выполнения программы:

